

**MC937A/MO603A – Computação Gráfica - 2021-S2 - Jorge Stolfi**  
**Trabalho de laboratório 07 - 2021-10-20**  
**Robôs com EA - O Filme (3)**

**Objetivos:** treinar animação de movimentos suaves com interpolação cúbica.

**Enunciado.** Nesta aula vamos terminar e aprimorar a simulação dos movimentos do robô, comparando interpolação afim (“linear”) com interpolação cúbica dos quadros-chave..

**Execução.** Você deve ter duas macros, digamos `quadro_inter1(tf)` e `quadro_inter3(tf)`, que devolve o robô, com as posições das juntas correspondentes a um instante arbitrário `tf` do filme; usando interpolação afim e interpolação cúbica de Bézier, respectivamente. Como antes, `tf=0` e `tf=1` devem produzir a posição do robô no quadro inicial. Nas duas macros, o valor de cada parâmetro do robô deve ser uma função contínua do tempo `tf`, e deve ser o mesmo valor para `tf=0` e `tf=1`. Além disso, na macro `quadro_inter3`, a derivada do valor de cada parâmetro deve ser uma função contínua de `tf` (isto é, o gráfico do parâmetro em função de `tf` não deve ter quinas), e deve ter o mesmo valor para `tf=0` e `tf=1`.

O objetivo é produzir um filme em que dois robôs executam aproximadamente os mesmos movimentos ao mesmo tempo, mas um deles gerado por `quadro_inter1` e o outro por `quadro_inter3`.

**Arquivos.** Copie os arquivos da aula passada para uma nova sub-pasta 2021-10-20 da pasta `mc937` no seu computador. Edite o arquivo `main.pov`, conforme solicitado acima. Execute o comando `make fast` numa shell para gerar a imagem, e use seu navegador (com `file:///home/.../main.gif`) para ver o gif animado.

**Exportação.** Não se esqueça de **exportar seu arquivo `main.pov` até as 21:00 para [http://students.ic.unicamp.br/~raSEU\\_RA/mc937-2021-2/2021-10-20/](http://students.ic.unicamp.br/~raSEU_RA/mc937-2021-2/2021-10-20/)**

**Comandos.** Os seguintes comandos de POV-Ray são relevantes para esta tarefa:

**Movimento contínuo por partes.** Um movimento cíclico suave não pode ser executado com uma única interpolação de parâmetros `interpola3`. Então é necessário usar várias interpolações, e calcular os valores nas emendas (“nós”), como segue:

```
#macro quadro_inter3(tf)
  // Quadros chaves exceto os nos:
  #local T0 = 0.00;
  #local P0z = ...
  #local P1a = ...
  #local T1 = 0.20;
  #local P1z = ...
  #local P2a = ...
  #local T2 = 0.50;
  #local P2z = ...
  #local P3a = ...
  #local T3 = 0.70;
  #local P3z = ...
  #local P4a = ...
  #local T4 = 1.00;

  #local P4z = P0z
  #local P0a = P4a

  // Quadros chaves dos nos:
  #local P0m = param_interpola1(T0, T0-(T4-T3)/3, T0+(T1-T0)/3, P0a, P0z)
  #local P1m = param_interpola1(T1, T1-(T1-T0)/3, T1+(T2-T1)/3, P1a, P1z)
  #local P2m = param_interpola1(T2, T2-(T2-T1)/3, T2+(T3-T2)/3, P2a, P2z)
  #local P3m = param_interpola1(T3, T3-(T3-T2)/3, T3+(T4-T3)/3, P3a, P3z)
  #local P4m = param_interpola1(T4, T4-(T4-T3)/3, T4+(T1-T0)/3, P4a, P4z)

  #if ((T0 <= tf) & (tf <= T1))
    #local P = param_interpola3(tf, T0,T1, P0m,P0z,P1a,P1m);
  #elseif ((T1 <= tf) & (tf <= T2))
    #local P = param_interpola3(tf, T1,T2, P1m,P1z,P2a,P2m);
  #elseif ((T2 <= tf) & (tf <= T3))
    #local P = param_interpola3(tf, T2,T3, P2m,P2z,P3a,P3m);
  #elseif ((T3 <= tf) & (tf <= T4))
    #local P = param_interpola3(tf, T3,T4, P3m,P3z,P4a,P4m);
  #end
  robo_vet(P)
#end
```

```

#macro interpola3(tt, tt0,tt3, vv0,vv1,vv2,vv3)
#local vv01 = interpola1(tt, tt0,tt3, vv0,vv1);
#local vv12 = interpola1(tt, tt0,tt3, vv1,vv2);
#local vv23 = interpola1(tt, tt0,tt3, vv2,vv3);
#local vv012 = interpola1(tt, tt0,tt3, vv01,vv12);
#local vv123 = interpola1(tt, tt0,tt3, vv12,vv23);
#local vv0123 = interpola1(tt, tt0,tt3, vv012,vv123);
vv0123
#end

```

Este algoritmo calcula um polinômio de grau 3 que começa no ponto (tt0,vv0) e termina no ponto (tt3,vv3). As derivadas no início e fim do intervalo são determinadas pelos valores vv1 e vv2.

